

Refine Search

Search Results -

Terms	Documents
L8 and (((read or load) near6 (issu\$3 or satisf\$6 or process\$3 or complet\$3) with (new near3 value)) same (value near4 (write or store)))	0

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Search:

L9

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Sunday, August 29, 2004 [Printable Copy](#) [Create Case](#)

<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>	<u>Set</u> <u>Name</u>
side by side			result se
DB=USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ			
<u>L9</u>	L8 and (((read or load) near6 (issu\$3 or satisf\$6 or process\$3 or complet\$3) with (new near3 value)) same (value near4 (write or store)))	0	<u>L9</u>
<u>L8</u>	((read or load) with (write or store) with (simultaneous\$3 or concurrent\$3 or ("same" time)))	3912	<u>L8</u>
DB=PGPB,USPT; PLUR=YES; OP=ADJ			
<u>L7</u>	L6 and (((read or load) near6 (issu\$3 or satisf\$6 or process\$3 or complet\$3) with (new near3 value)) same (value near4 (write or store)))	1	<u>L7</u>
<u>L6</u>	L5 and ((read or load) with (write or store) with (simultaneous\$3 or concurrent\$3 or ("same" time)))	345	<u>L6</u>
<u>L5</u>	L4 and ((read or load) near6 value)	805	<u>L5</u>
<u>L4</u>	L3 and ((write or store) near6 value)	1070	<u>L4</u>
<u>L3</u>	L2 and ((read or write or store or load) with (delay\$3 or stall\$3 or complet\$3 or process\$4 or issu\$4) with (first or priorit\$6))	2231	<u>L3</u>
<u>L2</u>	(read or load) with (write or store) with (conflict\$3 or colli\$4 or match\$4 or bottleneck or relat\$4 or correspond\$3) with (address or line or location)	6334	<u>L2</u>
<u>L1</u>	read near4 (request or command or operation)	101039	<u>L1</u>



Welcome to IEEE Xplore®

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

IEEE Enterprise

- Access the IEEE Enterprise File Cabinet

Print Format

Terms used **read or load sentence write or store sentence simultaneous or concurrent or same time** and **read or load near/6 issu** or **satisf** or **process** or **complet sentence new near/3 value paragraph value ne**

Sort results by 

Display results 

 [Save results to a Binder](#)

[Try an Advanced Search](#)

 [Search Tips](#)

[Try this search in The ACM](#)

[Open results in a new window](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

1 [Fast detection of communication patterns in distributed executions](#)

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collabora**

Full text available:  [pdf\(4.21 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are used to obtain a better understanding of the execution of the application. The visualization tool we use is Poco, developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial

2 [Translator writing systems](#)

Jerome Feldman, David Gries

February 1968 **Communications of the ACM**, Volume 11 Issue 2

Full text available:  [pdf\(4.47 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

A critical review of recent efforts to automate the writing of translators of programming languages is presented. The study of syntax and its application to translator writing are discussed in Section II. Various approaches to a postsyntactic (semantic) aspects of translator writing are discussed in Section III, and several related topics are presented.

Keywords: compiler compiler-compiler, generator, macroprocessor, meta-assembler, metacompiler, parse, syntactic analysis, syntax, syntax-directed, translator, translator writing system

3 [Parallel execution of prolog programs: a survey](#)

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4

Full text available:  [pdf\(1.95 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Since the early days of logic programming, researchers in the field realized the potential for exploitation of the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computation patterns makes parallel execution more difficult.

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

4 [The berkeley UNIX consultant project](#)

Robert Wilensky, David N. Chin, Marc Luria, James Martin, James Mayfield, Dekai Wu

December 1988 **Computational Linguistics**, Volume 14 Issue 4

Full text available:  [pdf\(4.41 MB\)](#)

 [Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

UC (UNIX Consultant) is an intelligent, natural language interface that allows naive users to learn about the system. UC was undertaken because the task was thought to be both a fertile domain for artificial intelligence and a useful application of AI work in planning, reasoning, natural language processing, and knowledge representation. The implementation of UC comprises the following components: a language analyzer, called ALANA, produces a

5 Pen computing: a technology overview and a vision

André Meyer

July 1995 **ACM SIGCHI Bulletin**, Volume 27 Issue 3

Full text available:  pdf(5.14 MB)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This work gives an overview of a new technology that is attracting growing interest in public as well as in itself. The visible difference from other technologies is in the use of a pen or pencil as the primary means of a user and a machine, picking up the familiar pen and paper interface metaphor. From this follows a set of will be analyzed and put into context with other emerging technologies and visions. Starting with a short his

6 A shared, segmented memory system for an object-oriented database

Mark F. Hornick, Stanley B. Zdonik

January 1987 **ACM Transactions on Information Systems (TOIS)**, Volume 5 Issue 1

Full text available:  pdf(2.05 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper describes the basic data model of an object-oriented database and the basic architecture of the it. In particular, a secondary storage segmentation scheme and a transaction-processing scheme are discuss segmentation scheme allows for arbitrary clustering of objects, including duplicates. The transaction scheme different sharing protocols ranging from those that enforce serializability to those that are nonserializable a communi ...

7 Cyc: toward programs with common sense

Douglas B. Lenat, R. V. Guha, Karen Pittman, Dexter Pratt, Mary Shepherd

August 1990 **Communications of the ACM**, Volume 33 Issue 8

Full text available:  pdf(3.98 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Cyc is a bold attempt to assemble a massive knowledge base (on the order of 108 axioms) spanning human knowledge. This article examines the need for such an undertaking and reviews the authors' efforts over the begin its construction. The methodology and history of the project are briefly discussed, followed by a more treatment of the current state of the representation language used (epistemological level), techniques for e

8 A tour through cedar

Warren Teitelman

March 1984 **Proceedings of the 7th international conference on Software engineering**

Full text available:  pdf(2.08 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

9 Compiling nested data-parallel programs for shared-memory multiprocessors

Siddhartha Chatterjee

July 1993 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 15 Issue 3

Full text available:  pdf(4.17 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

Keywords: compilers, data parallelism, shared-memory multiprocessors

10 An Unclever Time-Sharing System

Caxton C. Foster

January 1971 **ACM Computing Surveys (CSUR)**, Volume 3 Issue 1

Full text available:  pdf(1.85 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes the internal structure of a time-sharing system in some detail. This system is dedicated to remote access, and has a simple file structure. It is intended for use in a university type environment where short jobs that will profit from one- or two-second turnaround. Despite its simplicity, this system can serve

introduction to the problems encountered by the designers of any time-sharing system. Included are a disc
common ...

11 The FINITE STRING Newsletter: Abstracts of current literature

Computational Linguistics Staff
January 1987 **Computational Linguistics**, Volume 13 Issue 1-2

Full text available:  pdf(6.15 MB)  Publisher Site Additional Information: [full citation](#)

12 Interactive Editing Systems: Part II

Norman Meyrowitz, Andries van Dam
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3

Full text available:  pdf(9.17 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

13 Efficient passage ranking for document databases

Marcin Kaszkiel, Justin Zobel, Ron Sacks-Davis
October 1999 **ACM Transactions on Information Systems (TOIS)**, Volume 17 Issue 4

Full text available:  pdf(328.98 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Queries to text collections are resolved by ranking the documents in the collection and returning the highest to the user. An alternative retrieval method is to rank passages, that is, short fragments of documents, as it improves effectiveness and identifies relevant material in documents that are too large for users to consider a ranking of passages. This can considerably increase retrieval costs. In this article we explore alternative query ev

Keywords: inverted files, passage retrieval, query evaluation, text databases, text retrieval

14 A history of the Promis technology: an effective human interface

Jan Schultz
January 1986 **Proceedings of the ACM Conference on The history of personal workstations**

Full text available:  pdf(2.61 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Scientific computing systems for individuals were pioneered early at Hewlett-Packard, beginning with the 9100 Calculator in 1968. Extensions of this first machine were soon seen in Personal Peripherals, such as Printers and Plotters, and followed by Graphic CRT Displays. By early 1972, the Desktop unit had been augmented by the Pocket Calculator, the ground-breaking HP 35A. This paper traces the evolution of these machines to the pre

15 Human-computer interface development: concepts and systems for its management

H. Rex Hartson, Deborah Hix
March 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 1

Full text available:  pdf(7.97 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [r](#)

Human-computer interface management, from a computer science viewpoint, focuses on the process of developing human-computer interfaces, including their representation, design, implementation, execution, evaluation, and management. This survey presents important concepts of interface management: dialogue independence, structural mode, interactive tools, rapid prototyping, development methodologies, and control structures. *Dialogue independence* is a key concept in interface management, as it allows users to interact with the system in a natural and intuitive way, without being constrained by the system's internal structure. *Structural mode* refers to the way the system is organized and managed, including its data structures, algorithms, and control logic. *Interactive tools* are used to facilitate the development and maintenance of interfaces, such as graphical user interfaces (GUIs) and command-line interfaces (CLIs). *Rapid prototyping* is a technique for quickly creating prototypes of interfaces, allowing users to test and refine them before final implementation. *Development methodologies* are used to guide the process of developing interfaces, such as user-centered design and iterative design. *Control structures* refer to the way the system is controlled and managed, including its user interface, data storage, and communication protocols.

16 Automatic parsing for content analysis

Frederick J. Damerau
June 1970 **Communications of the ACM**, Volume 13 Issue 6

Full text available:  pdf(4.07 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Although automatic syntactic and semantic analysis is not yet possible for all of the unrestricted natural language applications, of which content analysis is one, do not have such a stringent coverage requirement. Preliminary results of the Harvard Syntactic Analyzer can produce correct and unambiguous identification of the subject and object in approximately half of the relevant occurrences. This provides a degree of coverage for content analysis which is

Keywords: content analysis, information retrieval, language analysis, natural language processing, parsing text processing

17 Design of a separable transition-diagram compiler

Melvin E. Conway

July 1963 **Communications of the ACM**, Volume 6 Issue 7

Full text available:  pdf(1.88 MB) Additional Information: [full citation](#), [references](#), [citations](#)

18 Explicit multi-threading (XMT) bridging models for instruction parallelism (extended abstract)

Uzi Vishkin, Shlomit Dascal, Efraim Berkovich, Joseph Nuzman

June 1998 **Proceedings of the tenth annual ACM symposium on Parallel algorithms and architecture**

Full text available:  pdf(1.71 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

19 Spoken dialogue technology: enabling the conversational user interface

March 2002 **ACM Computing Surveys (CSUR)**, Volume 34 Issue 1

Full text available:  pdf(987.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Spoken dialogue systems allow users to interact with computer-based applications such as databases and e-mail using natural spoken language. The origins of spoken dialogue systems can be traced back to Artificial Intelligence research in the 1950s concerned with developing conversational interfaces. However, it is only within the last decade of advances in speech technology, that large-scale working systems have been developed and, in some cases, commercialized ...

Keywords: Dialogue management, human computer interaction, language generation, language understanding, speech recognition, speech synthesis

20 File servers for network-based distributed systems

Liba Svobodova

December 1984 **ACM Computing Surveys (CSUR)**, Volume 16 Issue 4

Full text available:  pdf(4.23 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)